Midterm II Exam, Spring 2013 | Monday April 29th, 2013 | Exam time: 07:00-9:00 P.M.

Student's name:.. **ID:** .... **Section:** ....

Problem 1 (8 points)

For each of the question below, circle either T (for **True**) or F (for **False**). **No explanations** are needed. Incorrect answers or unanswered questions are worth zero points.

T F Given a graph $G = (V, E)$ with cost on edges and a set $S \subseteq V$, let (u, v) be an edge such that (u, v) is the minimum cost edge between any vertex in S and any vertex in V-S. Then, the minimum spanning tree of G must include the edge (u, v). (You may assume the costs on all edges are distinct, if needed).

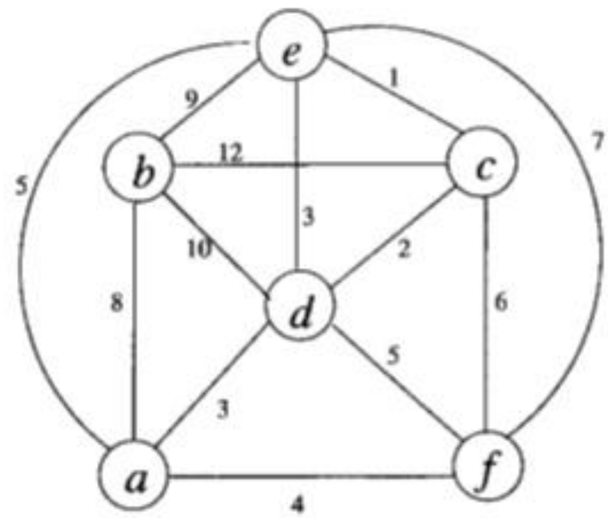
T F Let G be an edge-weighted directed graph with source vertex s and let T be a shortest path tree from s. Suppose we add a positive constant p to the cost of every edge in G. T remains a shortest path tree from s.

T F Let $G = (V, E)$ be a weighted graph and let M be a minimum spanning tree of G. The path in M between any pair of vertices v_1 and v_2 must be a shortest path in G

T F Consider a communication network of nodes where node v needs to broadcast a single message to all the other nodes efficiently. The message should be sent to the shortest paths tree from v.

Problem 2 (20 points)

For each of the algorithm below, list the edges of the Minimum Spanning Tree for the graph in the order selected by the algorithm.



a- Prim’s algorithm starting at vertex a.

b- Kruskal algorithm.

Problem 3 (9 points)

Compare dynamic programming, greedy programming, and standard recursion by filling out the table below

Algorithm	*Top-down or bottom-up?*	*Solve the same subproblem once?*	*Always solve all sub-problems*
Dynamic Programming			
Greedy Programming			
Standard Recursion			

Problem 4 (13 points)

Suppose **X**= cars and **Y**= cesar.

(a) Compute the length of an LCS of X and Y by filling out the c-matrix below

		c	**e**	**s**	**a**	**r**
c						
a						
r						
s						

(b) What is LCS of **X** and **Y**? For $\mathbf{X}_i$ and $\mathbf{Y}_j$ to be match in the LCS, what must be true about **c**[i,j]?

Problem 5 (10 points)

Give the pseudo-code of an algorithm that takes as input an array **A** of integers, and returns the length of the longest contiguous subsequence of odd numbers in **A**.

Example:

The length of the longest contiguous zeros subsequence in [1, 2, 19, 5, 4, 7, 51, 23, 22, 13, 15, 36] is 3.

What is the time complexity of your algorithm? hint: Use Dynamic programming paradigm.